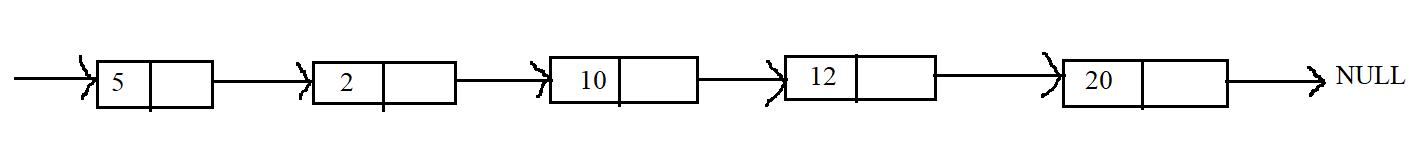
Searching a value into a created linked list:

Suppose a linked list is already created and your task is to find a value from that created linked list with its position. If the value is not belonging to the linked list then give a message.



Searching value is, y= any number.

Searching procedure:

Procedure SearchValueintoLinkedList

// create a linked list.

y searchingValue

p createdList

r p

while r (info) y and r NULL do

r r (next)

if r (info) y

print found

else

print Not Found

Inserting a node into a liked list:

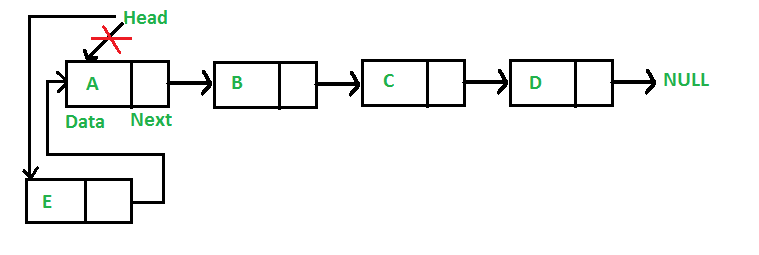
if we want to insert a node into linked list, we need to know the position that which position we need to insert the new node.

A node can be added in three ways

1. At the front of the linked list
2. After a given node.
3. At the end of the linked list.

Add a node at the front: (A 4 steps process)  
The new node is always added before the head of the given Linked List. And newly added node becomes the new head of the Linked List. For example, if the given Linked List is 10->15->20->25 and we add an item 5 at the front, then the Linked List becomes 5->10->15->20->25.

Let us call the function that adds at the front of the list is push (). The push () must receive a pointer to the head pointer, because push must change the head pointer to point to the new node



r = new node

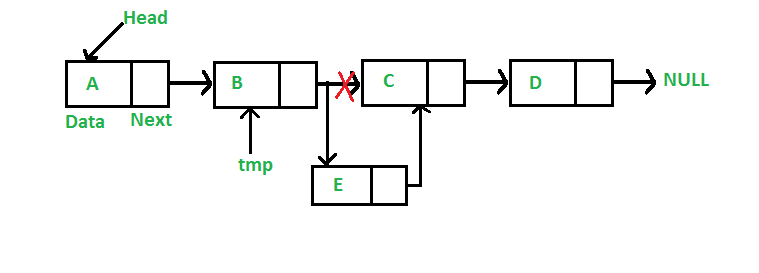
p = head of the created list

if (position == 1)

r -> next = p;

p = r;

Add a node after a given node: (5 steps process)  
We are given pointer to a node, and the new node is inserted after the given node.



p -> head of the created node

r = new node

place = 1;

q=p;

if (position ==3)

while (place != 2)

{

q = q->next;

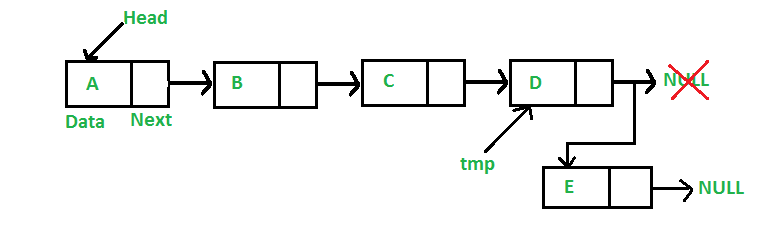
place++;

}

r - >next = q->next

q->next = r;

Add a node at the end: (6 steps process)  
The new node is always added after the last node of the given Linked List. For example if the given Linked List is 5->10->15->20->25 and we add an item 30 at the end, then the Linked List becomes 5->10->15->20->25->30.  
Since a Linked List is typically represented by the head of it, we have to traverse the list till end and then change the next of last node to new node.



q=p

while (q->next !=NULL)

{

q=q->next

}

q->next = r (5000)ss

r->next = NULL